



Python

数学实验与建模

司守奎 孙玺菁 主编

Python Mathematical
Experiment and Modeling



科学出版社

Python 数学实验与建模

司守奎 孙玺菁 主编

科学出版社

北京

内 容 简 介

本书以 Python 软件为基础,详细介绍了数学建模的各种常用算法及其软件实现,内容涉及高等数学、工程数学中的相关数学实验、数学规划、插值与拟合、微分方程、差分方程、评价预测、图论模型、多元分析、Monte Carlo 模拟、智能算法、时间序列分析、支持向量机、图像处理等内容,既有对算法数学原理的详述,又有案例和配套的 Python 程序. 本书含有 Python 快速入门基础,可以帮助 Python 零基础的读者快速掌握 Python 语言. 但对于没有其他任何编程语言基础的读者,建议参考一些更加具体的 Python 相关书籍.

本书可以作为本科生数学建模课程的主讲教材,也可以作为本科生数学实验课程的教材,以及运筹学课程的扩充阅读教材和教学参考书.

图书在版编目(CIP)数据

Python 数学实验与建模/司守奎, 孙玺菁主编. —北京: 科学出版社, 2020.4
ISBN 978-7-03-064527-2

I. ①P… II. ①司… ②孙… III. ①数学模型-软件工具-程序设计
IV. ①O141.4-39

中国版本图书馆 CIP 数据核字 (2020) 第 035397 号

责任编辑: 胡庆家 范培培 / 责任校对: 邹慧卿

责任印制: 吴兆东 / 封面设计: 无极书装

科学出版社 出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

北京虎彩文化传播有限公司 印刷

科学出版社发行 各地新华书店经销

*

2020 年 4 月第 一 版 开本: 720 × 1000 B5

2020 年 4 月第一次印刷 印张: 35 1/4

字数: 720 000

定价: 178.00 元

(如有印装质量问题, 我社负责调换)

编 写 组

主编：司守奎 孙玺菁

参编：司砚涵 张 原 刘 蕊 崔 晨

前 言

目前虽然有很多 Python 的相关书籍,但大多与人工智能、数据挖掘和金融方面等领域相关,涉及领域相对比较专一,同时很多书籍缺乏对模型或算法数学原理的详细阐述,理论基础相对薄弱,读者在阅读时往往会感觉不够深入.数学建模方面的相关书籍非常多,但多数书籍中的算法实现以 MATLAB 或 LINGO 语言为主,目前系统地将数学建模的各种模型和算法用 Python 实现的相关书籍还不多见.本书基于 Python 语言,对常用的数学模型和算法的数学理论进行阐述,同时结合案例实现 Python 的程序设计.

作者在编写本书之前,阅读了大量 Python 书籍.本书对 Python 在各个方面的编程技巧进行了归纳,以便读者从最方便的角度入手学习 Python.本书最大的特色在于:在对数学建模涉及的很多领域的模型及算法进行详细阐述的基础上,以 Python 为工具,以案例为主要模式,实现基于 Python 的程序设计.为广大希望以 Python 语言为基础、系统学习数学建模相关课程的读者提供了方便,也拓展了广大 Python 爱好者和使用者的学习广度.

本书总共 20 章,其中第 1 章 Python 语言快速入门、第 2 章数据处理与可视化、第 3 章 Python 在高等数学和线性代数中的应用、第 4 章概率论与数理统计、第 16 章 Monte Carlo 模拟、第 20 章数字图像处理可以作为数学实验教学的内容;其余章节和第 4 章作为数学建模教学的内容.在数学建模教学时,教师可以不讲授 Python 软件的具体内容,让学生把相关代码重新输入一遍.学习计算机语言必须输入一定量的代码,否则只能把他人的代码复制过来,进行简单的修改,没有真正学会编写算法的程序代码.

本书可以作为高等院校本科或研究生数学建模课程的教材或者参考书.本书是非常适合低起点 Python 初学者系统自学 Python 在数学建模中的程序设计的教材,也是适合具有一定 Python 基础的读者在数学和 Python 应用领域进行扩展的参考读物.

本书在编写过程中得到了海军航空大学王凤芹、海军工程大学刘海桥两位老师以及山东大学慕航同学的大力支持和帮助,作者对三位深表谢意.

最后,感谢科学出版社对本书出版所给予的大力支持,尤其是责任编辑胡庆家同志的热情支持与帮助.一本好的教材需要经过多年的教学实践,反复锤炼.由于

我们的经验和时间所限,书中的疏漏在所难免,敬请同行不吝指正.在使用过程中如果有问题,可以通过电子邮件和我们联系, E-mail: huqingjia@mail.sciencep.com, sishoukui@163.com, xijingsun1981@163.com, 也可以加入 QQ 群 554385668 和作者进行交流.

作 者

2019 年 8 月

目 录

前言	
第 1 章 Python 语言快速入门	1
1.1 Python 的安装与简单使用	1
1.1.1 Python 系统的安装	1
1.1.2 Python 工具库的管理与安装	3
1.1.3 简单的 Python 程序	4
1.2 Python 基础知识	5
1.2.1 基本数据处理	5
1.2.2 输出 print 和输入 input	6
1.2.3 运算符与表达式	8
1.2.4 流程控制	10
1.3 复合数据类型	14
1.3.1 list 列表	14
1.3.2 tuple 元组、dict 字典和 set 集合	17
1.3.3 序列的一些实用操作	20
1.4 函数	24
1.4.1 自定义函数语法	24
1.4.2 自定义函数的四种参数	25
1.4.3 参数传递	27
1.4.4 两个特殊函数	29
1.4.5 导入模块	31
1.5 Python 程序的书写规则	34
习题 1	36
第 2 章 数据处理与可视化	39
2.1 数值计算工具 NumPy	39
2.1.1 数组的创建、属性和操作	39
2.1.2 数组的运算、通用函数和广播运算	45
2.1.3 NumPy.random 模块的随机数生成	48
2.1.4 文本文件和二进制文件存取	48
2.2 文件操作	53

2.2.1	文件基本操作	53
2.2.2	文本文件的读写操作	55
2.2.3	文件管理方法	56
2.3	数据处理工具 Pandas	57
2.3.1	Pandas 的序列与数据框	58
2.3.2	外部文件的存取	60
2.4	Matplotlib 可视化	64
2.4.1	基础用法	65
2.4.2	Matplotlib.pyplot 的可视化应用	68
2.4.3	可视化的综合应用	74
2.5	scipy.stats 模块简介	78
2.5.1	随机变量及分布	78
2.5.2	概率密度函数和分布律可视化	79
	习题 2	83
第 3 章	Python 在高等数学和线性代数中的应用	85
3.1	SymPy 工具库介绍	85
3.1.1	PymPy 工具库简介	85
3.1.2	符号运算基础知识	87
3.2	SciPy 工具库简介	88
3.3	用 SymPy 做符号函数画图	91
3.4	高等数学问题的符号解	93
3.5	高等数学问题的数值解	98
3.5.1	泰勒级数与数值导数	98
3.5.2	数值积分	101
3.5.3	非线性方程(组)数值解	104
3.5.4	函数极值点的数值解	107
3.6	线性代数问题的符号解和数值解	108
3.6.1	线性代数问题的符号解	108
3.6.2	线性代数问题的数值解	113
3.6.3	求超定线性方程组的最小二乘解	118
	习题 3	120
第 4 章	概率论与数理统计	122
4.1	随机变量的概率计算和数字特征	122
4.1.1	随机变量的概率计算	122
4.1.2	随机变量数字特征简介	123

4.1.3 随机变量数字特征计算及应用	125
4.2 描述性统计和统计图	127
4.2.1 统计的基础知识	127
4.2.2 用 Python 计算统计量	129
4.2.3 统计图	132
4.3 参数估计和假设检验	140
4.3.1 参数估计	140
4.3.2 参数假设检验	142
4.3.3 非参数假设检验	146
4.4 方差分析	150
4.4.1 单因素方差分析及 Python 实现	151
4.4.2 双因素方差分析及 Python 实现	155
4.5 一元线性回归模型	160
4.5.1 一元线性回归分析	160
4.5.2 一元线性回归应用举例	164
4.6 常用的数据清洗方法	166
4.6.1 重复观测处理	167
4.6.2 缺失值处理	168
4.6.3 异常值处理	170
习题 4	173
第 5 章 线性规划	175
5.1 线性规划的概念和理论	175
5.2 线性规划的 Python 求解	177
5.2.1 用 scipy.optimize 模块求解	177
5.2.2 用 cvxopt.solvers 模块求解	182
5.2.3 用 cvxpy 求解	183
5.3 灵敏度分析	185
5.4 投资的收益和风险	187
习题 5	193
第 6 章 整数规划与非线性规划	195
6.1 整数规划	195
6.1.1 整数规划问题与求解	195
6.1.2 指派问题及求解	196
6.1.3 整数规划实例 —— 装箱问题	200
6.2 非线性规划	202

6.2.1	非线性规划概念和理论	202
6.2.2	非线性规划的 Python 求解	205
6.2.3	飞行管理问题	209
习题 6		213
第 7 章	插值与拟合	215
7.1	插值	215
7.1.1	插值方法	215
7.1.2	用 Python 求解插值问题	221
7.2	拟合	225
7.2.1	最小二乘拟合	225
7.2.2	数据拟合的 Python 实现	228
习题 7		231
第 8 章	微分方程模型	234
8.1	微分方程模型的求解方法	234
8.1.1	微分方程的数值解	234
8.1.2	用 Python 求解微分方程	235
8.2	微分方程建模方法	240
8.3	微分方程建模实例	245
8.3.1	Malthus 模型	245
8.3.2	Logistic 模型	246
8.3.3	美国人口的预报模型	247
8.3.4	传染病模型	249
8.4	拉氏变换求常微分方程 (组) 的符号解	252
习题 8		255
第 9 章	综合评价方法	257
9.1	综合评价的基本理论和数据预处理	257
9.1.1	综合评价的基本概念	257
9.1.2	综合评价体系的构建	258
9.1.3	评价指标的预处理方法	260
9.1.4	评价指标预处理示例	264
9.2	常用的综合评价数学模型	266
9.2.1	线性加权综合评价模型	266
9.2.2	TOPSIS 法	267
9.2.3	灰色关联度分析	268
9.2.4	熵值法	269

9.2.5 秩和比法	269
9.2.6 综合评价示例	271
9.3 层次分析法案例	274
习题 9	280
第 10 章 图论模型	281
10.1 图的基础理论及 networkx 简介	281
10.1.1 图的基本概念	281
10.1.2 图的表示及 networkx 简介	284
10.2 最短路算法及其 Python 实现	289
10.2.1 固定起点到其余各点的最短路算法	290
10.2.2 每对顶点间的最短路算法	293
10.2.3 最短路应用范例	297
10.3 最小生成树算法及其 networkx 实现	301
10.3.1 基本概念	301
10.3.2 求最小生成树的算法	302
10.3.3 用 networkx 求最小生成树及应用	304
10.4 匹配问题	306
10.5 最大流与最小费用流问题	309
10.5.1 最大流问题	309
10.5.2 最小费用流问题	312
10.6 PageRank 算法	314
10.7 复杂网络简介	318
10.7.1 复杂网络初步介绍	318
10.7.2 复杂网络的统计描述	319
习题 10	323
第 11 章 多元分析	326
11.1 判别分析	326
11.1.1 距离判别法	326
11.1.2 Fisher 判别法	330
11.1.3 贝叶斯判别法	332
11.1.4 判别准则的评价	333
11.2 主成分分析	335
11.2.1 主成分分析的基本原理和步骤	335
11.2.2 主成分分析的应用	339
11.3 因子分析	342

11.3.1	因子分析的数学理论	342
11.3.2	学生成绩的因子分析模型	346
11.4	聚类分析	350
11.4.1	数据变换	350
11.4.2	样品间亲疏程度的测度计算	351
11.4.3	scipy.cluster.hierarchy 模块的层次聚类	353
11.4.4	基于类间距离的层次聚类	355
11.4.5	K 均值聚类	358
11.4.6	K 均值聚类法最佳簇数 k 值的确定	360
11.4.7	K 均值聚类的应用	363
习题 11		366
第 12 章	回归分析	369
12.1	多元线性回归分析	369
12.1.1	多元线性回归模型	369
12.1.2	Python 求解线性回归分析	372
12.2	线性回归模型的正则化	374
12.2.1	多重共线性关系	375
12.2.2	岭回归	377
12.2.3	LASSO 回归	379
12.3	Logistic 回归	383
12.3.1	Logistic 回归模型	383
12.3.2	Logistic 回归模型的应用	387
习题 12		391
第 13 章	差分方程模型	394
13.1	差分方程及解法	394
13.2	差分方程的平衡点及稳定性	398
13.3	Leslie 模型	399
13.4	管住嘴迈开腿	404
13.5	离散阻滞增长模型及其应用	409
13.5.1	离散阻滞增长模型	409
13.5.2	离散阻滞增长模型的应用	411
13.6	染色体遗传模型	413
习题 13		416
第 14 章	模糊数学	418
14.1	模糊数学的基本概念和基本运算	418

14.1.1	模糊数学的基本概念	418
14.1.2	模糊数学的基本运算	421
14.2	模糊模式识别	424
14.2.1	择近原则	424
14.2.2	最大隶属原则	426
14.3	模糊聚类	427
14.3.1	模糊层次聚类	427
14.3.2	模糊 C 均值聚类	431
14.4	模糊综合评价	434
	习题 14	439
第 15 章	灰色系统预测	441
15.1	灰色系统理论简介	441
15.2	灰色 GM(1, 1) 预测模型	444
15.3	灰色 GM(1, N) 预测模型	449
15.4	灰色 GM(2, 1) 预测模型	453
	习题 15	457
第 16 章	Monte Carlo 模拟	458
16.1	随机变量的模拟	458
16.2	Monte Carlo 方法的数学基础及思想	461
16.3	随机模拟的应用	462
	习题 16	471
第 17 章	智能算法	472
17.1	模拟退火算法	472
17.1.1	模拟退火算法简介	472
17.1.2	算法流程及应用	473
17.2	遗传算法	478
17.2.1	遗传算法的原理	478
17.2.2	遗传算法应用	480
17.3	人工神经网络	483
17.3.1	人工神经网络概述	483
17.3.2	神经网络的基本模型	484
17.3.3	神经网络的应用	488
	习题 17	491
第 18 章	时间序列分析	493
18.1	移动平均法、指数平滑法和季节模型	493

18.1.1	移动平均法	493
18.1.2	指数平滑法	496
18.1.3	具有季节性时间序列的预测	500
18.2	平稳时间序列分析	502
18.2.1	基本概念和理论	502
18.2.2	ARMA 模型的构建及预报	505
18.2.3	ARMA 模型的 Python 求解	509
18.3	非平稳时间序列	513
	习题 18	516
第 19 章	支持向量机	518
19.1	支持向量分类机的基本原理	518
19.1.1	线性可分支持向量分类机	518
19.1.2	广义线性可分支持向量分类机	521
19.1.3	线性不可分支持向量分类机	523
19.2	支持向量回归	524
19.3	支持向量机的应用	526
19.3.1	支持向量机的分类问题	526
19.3.2	支持向量回归分析	528
	习题 19	530
第 20 章	数字图像处理	531
20.1	数字图像概述	531
20.1.1	图像的概念及表示	531
20.1.2	数字图像处理涉及的 Python 库	533
20.2	PIL 库的模块介绍	538
20.3	PIL 在安全领域的应用	542
20.3.1	生成验证码图片	542
20.3.2	给图像添加水印	544
20.3.3	生成二维码	545
20.3.4	拼图问题	547
	习题 20	548
	参考文献	549

第 1 章 Python 语言快速入门

Python 是一种可以撰写跨平台应用程序的解释型、面向对象的高级程序设计语言。Python 的设计哲学强调代码的可读性和语法的简洁性，尤其是使用空格缩进划分代码块，而非使用大括号或者关键词。让开发者能够用更少的代码表达想法，不管是小型还是大型程序，该语言都试图让程序的结构清晰明了。由于语法简洁而清晰，非常容易上手，且具有丰富和强大的类库，它往往能够用几行简单的代码就可以驱动操作系统及实现应用程序的多样化功能，因此它又常被称为胶水语言。

1.1 Python 的安装与简单使用

运行 Python 程序需要相应开发环境的支持。Python 内置的命令解释器（称为 Python Shell，Shell 有操作的接口或外壳之意）提供了 Python 的开发环境 IDLE（集成开发环境），能方便地进行交互式操作，即输入一行语句，就可以立刻执行该语句，并看到执行结果。此外，还可以利用第三方的 Python 集成开发环境进行程序设计。

1.1.1 Python 系统的安装

1. 安装 Python 基本库

要使用 Python 语言进行程序开发，必须安装其开发环境，即 Python 解释器。安装前先要从 Python 官网下载 Python 安装文件，下载地址为 <http://www.python.org/downloads>。选择基于 Windows 操作系统的 Python 3.7.2 进行下载，不要用最新版本，否则后面的凸优化库 `cvxpy` 是无法安装的。

下载完成后，运行文件 `python-3.7.2-amd64.exe`，进入 Python 系统安装界面，如图 1.1 所示。选中“Add Python 3.7 to PATH”复选框，并使用默认的安装路径，单击“Install Now”选项，这时进入系统安装过程，安装完成后单击“Close”按钮即可。如果要设置安装路径和其他特性，可以选择“Customize installation”。

2. 系统环境变量的设置

在 Python 的默认安装路径下包含 Python 的启动文件 `python.exe`、Python 库文件和其他文件。为了能在 Windows 命令提示符窗口自动寻找安装路径下的文件，需要将 Python 安装文件夹添加到环境变量 Path 中。



图 1.1 Python 安装示意图

如果在安装时选中了“Add Python 3.7 to PATH”复选框，则会自动将安装路径添加到环境变量 Path 中，否则可以在安装完成后添加，其方法为：在 Windows 桌面右击“计算机”图标，在弹出的快捷菜单中选中“属性”，然后在打开的对话框中选择“高级系统设置”选项，在打开的“系统属性”对话框中选择“高级”选项卡，单击“环境变量”按钮，打开“环境变量”对话框，在“系统变量”区域选择“Path”选项，单击“编辑”按钮，把安装路径添加到 Path 中，最后单击“确定”按钮逐级返回。

安装完 Python 3.7.2 后，实际上只安装了 Python 的基本库，Python 的 NumPy, SciPy, SymPy, Pandas 和 Matplotlib 等核心库都没有安装，建议初学 Python 者安装 Anaconda 开发环境。

3. Anaconda 开发环境

Anaconda (<https://www.anaconda.com/>) 是 Anaconda 公司提供的 Python 集成版。包括近 200 多个工具库，常见的库有 NumPy, SciPy, Pandas, IPython, Matplotlib, Scikit-learn 和 NLTK 等。它是一个跨平台的版本，可以与其他现有的 Python 版本一起安装。其基础版本是免费的，其他具有高级功能的附加组件需单独收费。Anaconda 自带库管理器 conda，通过命令行来管理安装库。

下载完 Anaconda3-2018.12-Windows-x86.64.exe 文件，运行该文件，进行“傻瓜式”安装即可。安装完 Anaconda 后，就可以使用其中的 Spyder 集成开发环境，Spyder 开发环境比 Python 自带的 IDLE 方便，Spyder 环境下表达式的值计算完成后，马上就可以看到其值；而 Python 自带的 IDLE 下，表达式值计算完成后，并不显示，需要用 print 语句显示其值。

1.1.2 Python 工具库的管理与安装

Python 有两个最主要的特征,一个是与其他语言相融合的能力,另一个是成熟的软件库系统.

1. 使用 pip 管理扩展库

目前, pip 已经成为管理 Python 扩展库的主流方式,大多数扩展库都支持这种方式进行安装、升级、卸载等操作,使用这种方式管理 Python 扩展库只需要在保证计算机联网的情况下输入几个命令即可完成,极大地方便了用户.

在 Python 3.4.0 之后的安装包中已经集成了 pip 工具,安装后的可执行文件在 Python37\Scripts\ 目录下. Python 3.4.0 之前的版本,需要另外安装 pip 工具,首先从 <https://pypi.org/project/pip/> 下载文件 get-pip.py,然后在命令提示符 (运行 cmd) 下执行命令

```
python get-pip.py
```

即可自动完成 pip 的安装.当然,需要保证计算机处于联网状态.

安装完成以后,就可以在命令提示符下使用 pip 来完成扩展库的安装、升级、卸载等操作, pip 常用命令的使用方法如表 1.1 所示.

表 1.1 pip 常用命令的使用方法

pip 命令示例	说明
pip install SomePackage	安装 SomePackage
pip list	列出当前已安装的所有库
pip install--upgrade SomePackage	升级 SomePackage 库
pip install-U SomePackage	
pip uninstall SomePackage	卸载 SomePackage 库

2. cvxpy 优化工具库安装

cvxpy 库的安装是一个很麻烦的过程,它不支持在线安装.网上有很多关于 cvxpy 库安装的各种报错信息的处理方式.读者只需要按照以下的步骤手动安装,就不会报错.以系统为 Windows 64 位 + Python 3.7 的计算机为例 (cvxpy 只支持 Python 3.7.2 版本,高版本的 Python 无法运行 cvxpy 库).

cvxpy 库所依赖的工具库有很多,有 NumPy+mkl, SciPy, cvxopt, scs, ecos, fastcache 和 osqp 等等,这些工具库下载到本地计算机中,并使用 1.2 类似的方式安装.需要注意的有两点,一是安装的工具库版本必须与 Python 版本和系统相对应,其中源文件名中的 cp37 表示 Python 3.7, amd64 表示 64 位, win32 表示 32 位.二是 NumPy 库的安装版本有很多,一定要选择 NumPy+mkl 库.